Remarks

Applicants respectfully request reconsideration of the present application in view of the foregoing amendments and the following remarks. Claims 1-28 are pending in the application. Claims 1-4, 9, 11, 19, 20, 24, and 25 are rejected. Claims 5-8, 10, 12-18, 21-23, and 26-28 are objected to. No claims have been allowed. Claims 1, 19, and 24 are independent. No claims have been amended.

Cited Art

The Action cites Parker (5,781,720) ("Parker").

Claim Rejections under 35 USC § 102

The Action rejects claims 1-4, 9, 11, 19-20, and 24-25 under 35 USC 102(b) as being anticipated by Parker. Applicants respectfully submit the claims are allowable over the cited art. For a 102(b) rejection to be proper, the cited art must show each and every element as set forth in a claim. (*See* MPEP § 2131.01.) However, the cited art does not describe each and every element. Accordingly, applicants request that all rejections be withdrawn. Claims 1, 19, and 24 are independent.

Claim 1 recites, in part:

summing costs assigned to data elements of the test input.

Claim 19 similarly recites, in part:

a module for summing costs assigned to data elements of the generated test input,

Claim 24 similarly recites, in part:

summing costs of data elements accesses.

For example, the Application describes examples summing costs of data elements for test inputs, along with motivations to do the same:

In one example, the technology restricts a potentially infinite search space for generating testing input. In one example, a tree generator constructs a tree of tree of data elements as an input to a program, such as a method, an API or a compiler. If the method is a compiler, the tree is language constructs that are parsed into a tree or graph data structure. Starting from the root of the graph or

tree inputs, a method assigns costs to the links of the graph. *A test developer sets a maximum cost as input to the method, and the costs of data elements are summed as the graph is built. In another example, a method monitors the predicate as it accesses data elements, and sums the costs of elements accessed by the predicate.* In one example, the method sums the costs of a path through the element. In another example, the method sums the costs for plural paths through the tree. In one example, the global bound is enforced by a module monitoring the execution of the predicate and recording data accesses made by the predicate. *As the data nodes of the tree are traversed by the predicate via the links, the global costs are summed for each node. The global bound on costs effectively allows restricting overall size of the inputs, in addition to, or instead of, restricting the number of instances of particular data types.*

[Application, at page 7, lines 5-20.]

  Parker does not teach or suggest "summing costs assigned to data elements" or "summing costs of data elements accesses" because it performs testing on every user event in its test script, regardless of cost. In its "Summary of the Invention," Parker describes its test script component:

>   The invention has essentially three major components. *The first component is a test script which is written in a high level programming language and contains the user events to be simulated, and the control and data structures necessary to validate the GUI's, and in turn, the application program's responses to the input.* The test script is preferably written in such a manner that it is directed towards operation on logical objects, instead of GUI-specific references. *The second component of the invention is a test executive. The test executive executes the test script.*

[Parker, at column 4, lines 6-15; emphasis added.] Parker thus describes its test executive as "execut[ing] the test script," which implies execution of all listed test events without regard to their costs. Parker also demonstrates execution of tests without regard to costs at Figure 6. As Parker notes, Figure 6 "*illustrates a high level flow for the calling of functions from a test script.*" [Parker, at column 15, lines 20-22.] As Figure 6, and its accompanying text, shows, every function in the test script is transmitted as input actions to the tested GUI by the process of Figure 6. [*See*, Parker, at column 15, lines 20-55.] The process of Figure 6 does not contain any provision for determining the cost of a function in Parker's test script, and furthermore has no need to, as it executes every function.

For at least these reasons, Parker cannot show "summing costs assigned to data elements" or "summing costs of data elements accesses." Furthermore, Applicants fail to find relevant disclosure in the passage of Parker cited in the Action which states:

> The testing methodology provided by the present invention allows the principles of software engineering to be applied to the testing of applications as well as to the development of such programs. The discipline of software engineering was developed primarily to make it possible to create large quantities of complex code in such a way that it can be managed and maintained at reasonable cost.

[Parker, at column 27, lines 26-32]. While Applicants note the use of the word "cost" in the cited passage, Applicants respectfully note that the "cost," as used in the passage, relates to "the principles of software engineering." It is related to the entire cost of developing software; hence the passage's reference to "creat[ing] large quantities of complex code." There is no indication in the passage that the "cost" is in any way related to "data elements" as are recited in the code. Furthermore, the cited passage does not discuss, teach or suggest the "summing" of costs related to "data elements," nor does any other passage in Parker, for that matter.

For at least these reasons, Parker does not teach or suggest every element of the independent claims 1, 19, and 24 and thus the claims are allowable over Parker. With respect to the dependent claims 2-18, 20-23, and 25-28, Applicants do not further belabor the individual patentability of these claims, but note that each depends from one of claims 1, 19, and 24 and recites patentability distinct subject matter. For at least these reasons claims 1-28 are allowable. Applicants request that the rejection of claims 1-4, 9, 11, 19-20, and 24-25 be withdrawn and that the claims be allowed.

Interview Request

If the claims are not found by the Examiner to be allowable, the Examiner is requested to call the undersigned attorney to set up an interview to discuss this application.
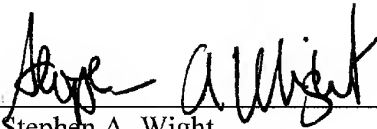
Conclusion

The claims in their present form should be allowable. Such action is respectfully requested.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Telephone: (503) 595-5300
Facsimile: (503) 595-5301

By
Stephen A. Wight
Registration No. 37,759